Theoretical Computer Science 131 (1994) 121–138
Elsevier

Realizability interpretation of generalized inductive definitions

Satoshi Kobayashi

Department of Applied Mathematics and Informatics, Ryukoku University, Seta, Ohtsu 520-21, Japan

Makoto Tatsuta

Research Institute of Electrical Communication, Tohoku University, Katahira, Sendai 980-77, Japan

Communicated by M. Nivat
Received January 1993
Revised October 1993

Abstract

Kobayashi, S. and M. Tatsuta, Realizability interpretation of generalized inductive definitions, Theoretical Computer Science 131 (1994) 121–138.

Generalized inductive definitions give a way of defining a predicate as the least solution P of the equation $P \leftrightarrow A[P]$ where a predicate variable P may occur in a formula $A[P]$ positively. This paper gives a **q**-realizability interpretation of generalized inductive definitions and proves the soundness of the interpretation.

1. Introduction

We study **q**-realizability of generalized inductive definitions to give a logical system in which we can formalize properties of programs by using generalized inductive definitions and extract a program from a constructive proof by **q**-realizability. This paper presents an improved version of the main result of [10].

In recent years, the idea of "proof-as-program" has drawn much attention in the area of computer science [2, 8]. The key of the idea is to make use of the existence property of a constructive formal system to produce programs. The existence property holds for a constructive formal system. When we prove an existence theorem

Correspondence to: S. Kobayashi, Department of Applied Mathematics and Informatics, Ryukoku University, Seta, Ohtsu 520-21, Japan. Email addresses of the authors: satoshi@rins.ryukoku.ac.jp and tatsuta@riec.tohoku.ac.jp.

SSDI 0304-3975(93)E0203-G

constructively, by the existence property, we can extract a program from the proof which computes the solution.

Various metamathematical devices can be used to extract programs. One possibility is to use a **q**-realizability interpretation or its variants. Actually, Hayashi and Nakano [7] adopted a variant of Grayson-type **q**-realizability in their system **PX** to extract a LISP program. Another possibility is to work in constructive type theories [4, 9].

We use realizability to extract programs from constructive proofs.

The inductive definition principle of predicates (or sets) is significant in computer science for the following reasons. Firstly, as is shown by the studies in logic programming and program specification, inductive definitions of predicates by Prolog-like Horn clauses are very useful in program specification. The generalized inductive definitions include the inductive definitions by Horn clauses. Secondly, many important data types such as integers, lists, trees, etc., are inductively defined. Thirdly, as Hayashi and Nakano [7] pointed out, inductive definitions play essentially important roles in extraction of efficient programs from constructive proofs.

For these reasons, logical systems to formalize properties of programs such as program extraction systems should admit a sufficiently broad class of inductive definitions.

Feferman [5, 6] introduced the constructive theory T_0 of functions and classes, and defined some forms of realizability for it. T_0 has axioms for accessibility inductive definition of classes. Buchholz [3] deals with recursive **r**-realizability for strictly positive $\mathbf{ID}^i_\nu$, which is a theory of ν-times iterated strictly positive inductive definitions. In Hayashi and Nakano's **PX** [7], the inductive definition principle called CIG (conditional inductive generation) is available. Operator forms of CIG are restricted to positive rank 0 forms. (Roughly speaking, "rank 0" means "Harrop".)

Tatsuta [10] deals with a more general case than these works. In his system, an inductive operator form $A[X]$ must be positive and must satisfy the following side condition: every occurrence of X in $A[X]$ must be either strictly positive or contained in a Harrop subformula of $A[X]$. He defined the **q**-realizability interpretation for this class of inductive definitions and proved its soundness. The disadvantage of the work is that the side condition seems quite artificial.

This paper is based on [10] and improves the theoretical results given in [10]. In this paper, we define a **q**-realizability interpretation of *arbitrary* positive inductive definitions (with no side conditions) and prove the soundness of the interpretation. We show that the artificial side condition given in [10] is superfluous and can be removed.

By the improved result given by this paper, our system can use nonstrictly positive inductive definitions, which the system given in [10] cannot cover. Therefore our system can be used to extract programs which treat nonstrictly-positive inductive datatypes. In particular, our system can formalize continuations, whose datatype *cont* is given by $cont = 1 + ((cont \to \alpha) \to \alpha)$.

Higher-order codings of higher-order systems such as the second-order predicate logic $\mathbf{PA}_2$ also give the generalized inductive definitions. By the well-known fact,

q-realizability for the system $\mathbf{PA}_2$ gives monotone inductive definitions. We do not choose the framework of higher-order codings for the following reasons. Firstly, those systems which have the facility of higher-order codings are impredicative and too strong to be used in program extraction. Secondly, induction schemata given by higher-order codings do not give loop structures to programs since $(e\,\mathbf{q}\,\mu)\leftrightarrow(e\,\mathbf{q}\,A[\mu])$ does not hold where $\mu\equiv\mu X.A[X]$.

The definition of our realizability is essentially due to Tatsuta [10]. The key lemma and the main result were proved by Kobayashi.

Section 2 presents the theory $\mathbf{EON}+\mu$. Section 3 defines the **q**-realizability interpretation for $\mathbf{EON}+\mu$. Section 4 proves the soundness theorem of the interpretation.

2. Language and axioms of EON + μ

In this section, we introduce the formal system $\mathbf{EON}+\mu$, which is a theory of finitely iterated inductive definitions. This system is an extension of Beeson's formal system **EON** [1]. To keep the argument as abstract as possible, we chose **EON** as the base theory rather than Heyting's arithmetic. Our system is also an extension of the original version of $\mathbf{EON}+\mu$ described in Tatsuta [10].

2.1. Language. (1) Constant symbols: $\mathbf{k}, \mathbf{s}, \mathbf{p}, \mathbf{p}_0, \mathbf{p}_1, \mathbf{0}, \mathbf{s}_N, \mathbf{p}_N$ and $\mathbf{d}$.

(2) Function symbol: **Ap**.

(3) Predicate symbols: $\bot, \downarrow, N$ and $=$.

(4) Predicate variables: $P, Q, \ldots$. Each predicate variable has its arity. An arity is a nonnegative integer. For each nonnegative integer n, we have countably many predicate variables of arity n.

We use combinators as the target programming language. **k** and **s** are usual basic combinators. We have pairs as primitive by $\mathbf{p}, \mathbf{p}_0$ ad $\mathbf{p}_1$ which correspond to `cons`, `car` and `cdr` in LISP respectively. We have natural numbers as primitive by the zero 0, the successor $\mathbf{s}_N$ and the predecessor $\mathbf{p}_N$. **d** is a combinator judging the equality of natural numbers and corresponds to an if-then-else statement in usual programming languages. **Ap** means the functional application. $\bot$ means the contradiction. $a\downarrow$ means that a term a has a value. $N(a)$ means that a term a is a natural number. $a=b$ means that a term a is equal to a term b. A predicate variable is quantified only by a μ constructor and is not quantified by $\forall$ and $\exists$.

2.2. Definition (*Term*). (1) Variables and constants are terms.

(2) If t and s are terms, then $\mathbf{Ap}(t,s)$ is a term.

We write ts instead of $\mathbf{Ap}(t,s)$. $\mathbf{p}ts$ is usually written as $\langle t,s\rangle$.

2.3. Definition (*E-formula*). We define *E-formulae* ("*E*" stands for "extended") and their positive (negative) free predicate variables $S_+(A)$ ($S_-(A)$ resp.) simultaneously as follows.

(1) $\perp$ is an atomic E-formula. $S_+(\perp)=S_-(\perp)=\emptyset$.

(2) $t\downarrow$ is an atomic E-formula if t is a term. $S_+(t\downarrow)=S_-(t\downarrow)=\emptyset$.

(3) $N(t)$ is an atomic E-formula if t is a term. $S_+(N(t))=S_-(N(t))=\emptyset$.

(4) $t=s$ is an atomic E-formula if t and s are terms.

$$S_+(t=s)=S_-(t=s)=\emptyset.$$

(5) $P(\bar{t})$ is an atomic E-formula if P is a predicate variable of arity n and $\bar{t}$ is a list of terms of length n. $S_+(P(\bar{t}))=\{P\}$, $S_-(P(\bar{t}))=\emptyset$.

(6) $A \,\&\, B$ is an E-formula if A and B are E-formulae.

$$S_+(A \,\&\, B)=S_+(A)\cup S_+(B), \qquad S_-(A \,\&\, B)=S_-(A)\cup S_-(B).$$

(7) $A \vee B$ is an E-formula if A and B are E-formulae.

$$S_+(A \vee B)=S_+(A)\cup S_+(B), \qquad S_-(A \vee B)=S_-(A)\cup S_-(B).$$

(8) $A \rightarrow B$ is an E-formula if A and B are E-formulae.

$$S_+(A \rightarrow B)=S_-(A)\cup S_+(B), \qquad S_-(A \rightarrow B)=S_+(A)\cup S_-(B).$$

(9) $\forall x A(x)$ is an E-formula if x is a variable and $A(x)$ is an E-formula.

$$S_+(\forall x A(x))=S_+(A(x)), \qquad S_-(\forall x A(x))=S_-(A(x)).$$

(10) $\exists x A(x)$ is an E-formula if x is a variable and $A(x)$ is an E-formula.

$$S_+(\exists x A(x))=S_+(A(x)), \qquad S_-(\exists x A(x))=S_-(A(x)).$$

(11) $((\bar{x}).A)(\bar{t})$ is an E-formula if A is an E-formula, $\bar{x}$ is a list of variables of length n and $\bar{t}$ is a list of terms of length n.

$$S_+(((\bar{x}).A)(\bar{t}))=S_+(A), \qquad S_-(((\bar{x}).A)(\bar{t}))=S_-(A).$$

(12) $(\mu P.(\bar{x}).A)(\bar{t})$ is an E-formula if P is a predicate variable of arity n, $\bar{x}$ is a list of variables of length $n, \bar{t}$ is a list of terms of length n and P is not in $S_-(A)$.

$$S_+((\mu P.(\bar{x}).A)(\bar{t}))=S_+(A)-\{P\}, \qquad S_-((\mu P.(\bar{x}).A)(\bar{t}))=S_-(A).$$

The intended meaning of $\mu P.(\bar{x}).A[P]$ is the least predicate P such that $\forall \bar{x}(P(\bar{x})\leftrightarrow A[P])$.

2.4. Notation. A symbol with an overbar denotes a list of expressions. For example, we often use $\bar{t}$ to denote a list of terms. $\bar{t}\downarrow$ denotes $t_1\downarrow \& \ldots \& t_n\downarrow$.

We use the following abbreviations: $\neg A$ denotes $(A\rightarrow\perp)$ for an E-formula A. $a\neq b$ denotes $(a=b\rightarrow\perp)$ for terms a and b.

2.5. Definition. (1) P *occurs freely* in A if P is in $S_+(A)$ or in $S_-(A)$.

(2) A free predicate variable P in A are bound by μ in $(\mu P.(\bar{x}).A)(\bar{t})$.

(3) P is *positive* (*negative*) in A if P is not in $S_-(A)$ ($S_+(A)$, respectively).

In the following, we consider only such E-formula A that $S_+(A) \cap S_-(A) = \emptyset$. The set of free predicate variables $FPV(A)$ of A is defined by $FPV(A) = S_+(A) \cup S_-(A)$.

2.6. Definition (*E-predicate*). (1) Predicate symbols $\perp$, $\downarrow$, N and $=$ are E-predicates with arity $0, 1, 1$ and 2 respectively.

(2) A predicate variable P with arity n is an E-predicate with arity n.

(3) $\mu P.(\bar{x}).A$ is an E-predicate with arity n if $(\mu P.(\bar{x}).A)(\bar{x})$ is an E-formula and the length of $\bar{x}$ is n.

(4) $(\bar{x}).A$ is an E-predicate with arity n if A is an E-formula and the length of $\bar{x}$ is n.

We write *arity*(A) to denote the arity of E-predicate A. Free predicate variables $FPV(F)$ of an E-predicate F are defined in the same way as an E-formula.

2.7. Definition (*Formula, predicate*). A *formula* (*predicate*) is an E-formula (E-predicate, resp.) with no free occurrences of predicate variables.

2.8. Definition (*Substitution*). Substitution of terms for variables is defined as usual. For terms t and u and a variable x, $t[u/x]$ denotes a term obtained from t by replacing all the free occurrences of x by u. Substitution of predicates for predicate variables in a formula (or a predicate) is also defined as usual. For a formula A, a predicate variable P and a predicate F, $A_P[F]$ denotes a formula obtained from A by replacing all the free occurrences of P by F.

2.9. Logical rules and axioms. The underlying logic of our theory is Beeson's formal system **LPT** (logic of partial terms). It is a variant of first order intuitionistic predicate calculus. The propositional axioms and rules of inference are as usual.

The quanitifier axioms and rules are as follows.

(Q1) $\dfrac{B \to A}{B \to \forall x A}$ (x not free in B),

(Q2) $\dfrac{A \to B}{\exists x A \to B}$ (x not free in B),

(Q3) $\forall x A \,\&\, t\downarrow \to A[t/x]$,

(Q4) $A[t/x] \,\&\, t\downarrow \to \exists x A$.

The equality axioms are as follows:

(E1) $x=x\,\&\,(x=y\to y=x)$,

(E2) $t\simeq s\,\&\,\phi(t)\to\phi(s)$,

(E3) $t=s\to t\downarrow\&\,s\downarrow$.

Here t and s are terms, x and y are variables, and $t\simeq s$ is an abbreviation of $t\downarrow\vee s\downarrow\to t=s$.

The axioms of definedness are as follows.

(S1) $x\downarrow$ (every variable x),

(S2) $c\downarrow$ (every constant symbol c),

(S3) $ts\downarrow\to t\downarrow\&\,s\downarrow$,

(S4) $N(t)\to t\downarrow$,

(S5) $P(\bar{t})-\bar{t}\downarrow$,

(S6) $(\mu P.(\bar{x}).A)(\bar{t})\to\bar{t}\downarrow$.

In (S3) and (S4), t and s are terms. In (S5) and (S6), P is a predicate variable and $\bar{t}$ is a list of terms.

For details, see Beeson [1, Ch. VI].

2.10. Nonlogical axioms. Nonlogical axioms of our theory are those of Beeson's **EON**, axioms of abstracts and axioms of inductive definitions stated below.

Axioms of **EON**

(EON1) **k** *combinator*: $\mathbf{k}xy=x$.

(EON2) **s** *combinator*: $\mathbf{s}xyz\simeq xz(yz)\,\&\,\mathbf{s}xy\downarrow$.

(EON3) $\mathbf{k}\neq\mathbf{s}$.

(EON4) *Pairing*: $\mathbf{p}xy\downarrow\&\,\mathbf{p}_0(\mathbf{p}xy)=x\,\&\,\mathbf{p}_1(\mathbf{p}xy)=y$.

(EON5), (EON6) *Natural numbers*:

$$N(0)\,\&\,\forall x(N(x)\to[N(\mathbf{s}_N(x))\,\&\,\mathbf{p}_N(\mathbf{s}_Nx)=x\,\&\,\mathbf{s}_Nx\neq0]),$$

$$\forall x(N(x)\,\&\,x\neq0\to N(\mathbf{p}_Nx)\,\&\,\mathbf{s}_N(\mathbf{p}_Nx)=x).$$

(EON7), (EON8) *Definition by integer cases*:

$$N(a)\,\&\,N(b)\,\&\,a=b\to\mathbf{d}(a,b,x,y)=x,$$

$$N(a)\,\&\,N(b)\,\&\,a\neq b\to\mathbf{d}(a,b,x,y)=y.$$

(EON9) *Induction for all formulae*:

$$\phi(0) \,\&\, \forall x(N(x) \,\&\, \phi(x) \to \phi(\mathbf{s}_N x)) \to \forall x(N(x) \to \phi(x)).$$

Axioms of abstracts

(A1) $((\bar{x}).A)(\bar{t}) \leftrightarrow \bar{t}\downarrow \,\&\, A[\bar{t}/\bar{x}]$.

Axioms of inductive definitions

Let $\mu P.(\bar{x}).A[P]$ and C be predicates of the same arity. Let $B \equiv \mu P.(\bar{x}).A[P]$. Then the following (R1), (R2) are axioms:

(R1) $A[B] \to B(\bar{x})$,

(R2) $\forall \bar{x}(A[C] \to C(\bar{x})) \to \forall \bar{x}(B(\bar{x}) \to C(\bar{x}))$.

The consistency of the theory can be proved in the same way as [10].

2.11. Lemma. *Let* $B \equiv \mu P.(\bar{x}).A[P]$, *then we have* $\forall \bar{x}(B(\bar{x}) \leftrightarrow A[B])$.

This lemma is proved in [10].

2.12. Bracket abstraction. A combinatory term $\bar{t}$ is defined for a λ-term t in the following way.

(1) $\bar{x} \equiv x$ if x is a variable,
(2) $\bar{c} \equiv c$ if c is a constant,
(3) $\overline{st} \equiv \bar{s}\bar{t}$,
(4) $\overline{\lambda x.x} \equiv \mathbf{skk}$,
(5) $\overline{\lambda x.y} \equiv \mathbf{k}y$ if y is a variable and $x \not\equiv y$,
(6) $\overline{\lambda x.c} \equiv \mathbf{k}c$ if c is a constant,
(7) $\overline{\lambda x.st} \equiv \mathbf{s}(\overline{\lambda x.s})(\overline{\lambda x.t})$.

In the sequel, a term t always denotes a combinatory term $\bar{t}$ except that a substituted term $u[t/x]$ denotes a combinatory term $\bar{u}[\bar{t}/x]$ for terms t, u and a variable x.

Then

$$\lambda x.t\downarrow, \quad (\lambda x.t)x \simeq t$$

hold for a variable x and a term t.

2.13. The recursion theorem. Let $u_n \equiv \lambda y x_1 \ldots x_n.f(yy)x_1 \ldots x_n$ and $\mathbf{R}_n \equiv \lambda f.u_n u_n$. Then we have

$$\forall f \forall x_1, \ldots, x_n(\mathbf{R}_n f x_1 \ldots x_{n-1}\downarrow \,\&\, \mathbf{R}_n f x_1 \ldots x_n \simeq f(\mathbf{R}_n f)x_1 \ldots x_n).$$

Suppose f is a variable and $t[f]$ is a term possibly containing f. We write $\mu_n f.t[f]$ for $\mathbf{R}_n(\lambda f.t[f])$. Then we have

$$g=\mu_n f.t[f]\to\forall x_1,\ldots,x_n(gx_1\ldots x_{n-1}\downarrow \& \, gx_1\ldots x_n\simeq t[g]x_1\ldots x_n).$$

Sometimes we omit the subscript n of $\mu_n f.t$.

3. Realizability interpretation

3.1. Definition (*Extended* $\mathbf{q}$*-realizability Interpretation*).

For an E-formula A, a fresh variable e, predicate variables $\bar{P}\equiv P_1,\ldots,P_n$, predicates $\bar{F}\equiv F_1,\ldots,F_n$ and $\bar{G}\equiv G_1,\ldots,G_n$ such that $arity(F_i)=arity(P_i)$ and $arity(G_i)=arity(P_i)+1$, we define an E-formula

$$e\ \mathbf{q}_{\bar{P}}[\bar{F};\bar{G}]\ A$$

by induction on the construction of A. If $n=0$, we write $e\,\mathbf{q}\,A$ for it. We abbreviate

$$\mathbf{q}_{\bar{P}}[\bar{F};\bar{G}]$$

as $\mathbf{q}'$ and

$$\mathbf{q}_{\bar{P},Q}[\bar{F},H;\bar{G},J]$$

as $\mathbf{q}'_Q[H;J]$. Generally, if $\mathbf{q}'$ is an abbreviation of $\mathbf{q}_{\bar{P}}[\bar{F};\bar{G}]$, then $\mathbf{q}'_{\bar{Q}}[\bar{H};\bar{J}]$ is an abbreviation of $\mathbf{q}_{\bar{P},\bar{Q}}[\bar{F},\bar{H};\bar{G},\bar{J}]$. In the following, we write $A_{\bar{P}}[\bar{F}]$ for the result of simultaneous substitution of $\bar{F}$ for $\bar{P}$ in A. The definition is as follows.

(1) $e\,\mathbf{q}'\,P_i(\bar{t})\equiv G_i(e,\bar{t})$.

(2) $e\,\mathbf{q}'\,A\equiv e=0\,\&\,A$ if A is an atomic E-formula which is not of the form $P_i(\bar{t})$.

(3) $e\,\mathbf{q}'\,A\,\&\,B\equiv(\mathbf{p}_0 e\,\mathbf{q}'\,A)\,\&\,(\mathbf{p}_1 e\,\mathbf{q}'\,B)$.

(4) $e\,\mathbf{q}'\,A\vee B\equiv N(\mathbf{p}_0 e)\,\&\,(\mathbf{p}_0 e=0\to A_{\bar{P}}[\bar{F}]\,\&\,(\mathbf{p}_1 e\,\mathbf{q}'A))$
$\&\,(\mathbf{p}_0 e\neq 0\to B_{\bar{P}}[\bar{F}]\,\&\,(\mathbf{p}_1 e\,\mathbf{q}'\,B))$.

(5) $e\,\mathbf{q}'\,A\to B\equiv e\downarrow\,\&\,\forall q(A_{\bar{P}}[\bar{F}]\,\&\,(q\,\mathbf{q}'\,A)\to(eq\,\mathbf{q}'\,B))$.

(6) $e\,\mathbf{q}'\,\forall xA(x)\equiv\forall x(ex\,\mathbf{q}'\,A(x))$.

(7) $e\,\mathbf{q}'\,\exists xA(x)\equiv\mathbf{p}_0 e\downarrow\,\&\,A(\mathbf{p}_0 e)_{\bar{P}}[\bar{F}]\,\&\,(\mathbf{p}_1 e\,\mathbf{q}'\,A(\mathbf{p}_0 e))$.

(8) $e\,\mathbf{q}'((\bar{x}).A)(\bar{t})\equiv((r,\bar{x}).(r\,\mathbf{q}'\,A))(e,\bar{t})$.

(9) $e\,\mathbf{q}'(\mu Q.(\bar{x}).A[Q])(\bar{t})\equiv(\mu Q^*.(e,\bar{x}).(e\,\mathbf{q}'_Q[B_{\bar{P}}[\bar{F}];Q^*]\ A[Q]))(e,\bar{t})$,

where B is $\mu Q.(\bar{x}).A[Q]$ and Q^* is a fresh predicate variable whose arity is $arity(Q)+1$.

It is easy to see that the above definition is well-defined. The point is that if Q is positive (negative) in $A[Q]$, then Q^* is positive (negative, resp.) in $e\,\mathbf{q}'_Q[B,Q^*]\,A[P]$.

3.2. Remark. The notation $(e\ \mathbf{q}_{P_1 \ldots P_n}[F_1, \ldots, F_n; G_1, \ldots, G_n]\ A)$ used in this paper differs slightly from the notation used in [10]. The expression $(e\ \mathbf{q}_{P_1 \ldots P_n}[F_1, G_1, \ldots, F_n, G_n]\ A)$ used in [10] denotes the same formula as the expression $(e\ \mathbf{q}_{P_1 \ldots P_n}\ [F_1, \ldots, F_n; G_1, \ldots, G_n]\ A)$ used in this paper.

3.3. Lemma. (i) (*Substitution Property*)

$$(e\,\mathbf{q}_{\bar{P}}[\bar{F};\bar{G}]\,A)[y/x] \;\equiv\; e[y/x]\,\mathbf{q}_{\bar{P}}[\bar{F}[y/x];\bar{G}[y/x]]\,A[y/x].$$

Here $\bar{F}[y/x]$ *is* $F_1[y/x], \ldots, F_n[y/x]$.

(ii) *Suppose* A *is an E-formula and* $FPV(A) \subseteq \{\bar{P}\}$. *Then the free predicate variables of* $e\,\mathbf{q}_{\bar{P}}[\bar{F};\bar{G}]\,A$ *are among those of* $\bar{F}, \bar{G}$.

(iii) *If* A *is a formula, then* $e\,\mathbf{q}\,A$ *is also a formula.*

(iv) $(e\,\mathbf{q}_{\bar{P}}[\bar{F};\bar{G}]\,A) \to e\downarrow$.

These claims are proved in [10].

4. Soundness theorem

Now we shall show the soundness of our $\mathbf{q}$-realizability interpretation. The soundness for the axioms and rules of **EON** is proved in the same way as [1]. We prove that the axioms (A1), (R1) and (R2) are realizable.

4.1. Lemma. $\langle \lambda r.\langle 0, r\rangle, \lambda r.\mathbf{p}_1 r\rangle$ *realizes the axiom* (A1).

This lemma is proved immediately by the definition of realizability.

The following three lemmas are proved in [10].

4.2. Lemma. *Suppose* $\forall \bar{x}(H_1(\bar{x}) \to H_2(\bar{x}))$. *Then,*

(1) *if* P *is positive in* $A[\bar{R}, P]$, *we have*

$$(e\,\mathbf{q}_{\bar{R},P}[\bar{F}, H_1; \bar{G}, J]\ A[\bar{R}, P]) \;\to\; (e\,\mathbf{q}_{\bar{R},P}[\bar{F}, H_2; \bar{G}, J]\ A[\bar{R}, P]),$$

(2) *if* P *is negative in* $A[\bar{R}, P]$, *we have*

$$(e\,\mathbf{q}_{\bar{R},P}[\bar{F}, H_2; \bar{G}, J]\ A[\bar{R}, P]) \;\to\; (e\,\mathbf{q}_{\bar{R},P}[\bar{F}, H_1; \bar{G}, J]\ A[\bar{R}, P]).$$

4.3. Lemma. *We have*

$$e\,\mathbf{q}'_P[B; (e, \bar{x}).e\,\mathbf{q}'\,B(\bar{x})]\ A[\bar{R}, P] \;\equiv\; e\,\mathbf{q}'\,A[\bar{R}, B],$$

where $\mathbf{q}'$ *is an abbreviation of* $\mathbf{q}_{\bar{R}}[\bar{F};\bar{G}]$.

4.4. Lemma. *Let* $B \equiv \mu P.(\bar{x}).A[P]$. *Then,*

$$(e\,\mathbf{q}_{\bar{R}}[\bar{F};\bar{G}]B(\bar{x})) \leftrightarrow (e\,\mathbf{q}_{\bar{R}}[\bar{F};\bar{G}]A[B])$$

holds.

4.5. Corollary. $\lambda x.x$ *realizes the axiom* (R1)

4.6. Definition. For an E-formula A, a predicate variable P and a term f, we shall define a term $\sigma_A^{P,f}$ as follows:

(1) If $A \equiv P(\bar{t})$, then $\sigma_A^{P,f} \equiv \lambda r.f\bar{t}r$.

(2) If A is an atomic E-formula which is not of the form $P(\bar{t})$, then $\sigma_A^{P,f} \equiv \lambda r.r$.

(3) If $A \equiv A_1 \,\&\, A_2$, then $\sigma_A^{P,f} \equiv \lambda r.\langle \sigma_{A_1}^{P,f}(\mathbf{p}_0 r), \sigma_{A_2}^{P,f}(\mathbf{p}_1 r)\rangle$.

(4) If $A \equiv A_1 \vee A_2$, then $\sigma_A^{P,f} \equiv \lambda r.\langle \mathbf{p}_0 r, \mathbf{d}(\mathbf{p}_0 r)0\sigma_{A_1}^{P,f}\sigma_{A_2}^{P,f}(\mathbf{p}_1 r)\rangle$.

(5) If $A \equiv A_1 \rightarrow A_2$, then $\sigma_A^{P,f} \equiv \lambda rq.\sigma_{A_2}^{P,f}(r(\sigma_{A_1}^{P,f}q))$.

(6) If $A \equiv \forall x A_1(x)$, then $\sigma_A^{P,f} \equiv \lambda rx.\sigma_{A_1(x)}^{P,f}(rx)$.

(7) If $A \equiv \exists x A_1(x)$, then $\sigma_A^{P,f} \equiv \lambda r.\langle \mathbf{p}_0 r, \sigma_{A_1(\mathbf{p}_0 r)}^{P,f}(\mathbf{p}_1 r)\rangle$.

(8) If $A \equiv ((\bar{x}).A_1)(\bar{t})$, then $\sigma_A^{P,f} \equiv \lambda r.r$.

(9) If $A \equiv (\mu Q.(\bar{y}).A_1[P,Q])(\bar{t})$, then $\sigma_A^{P,f} \equiv g\bar{t}$, where g is defined as $g \equiv \mu_{m+1} g.\lambda \bar{y}r.\sigma_{A_1}^{Q,g}(\sigma_{A_1}^{P,f} r)$ and m is the length of $\bar{y}$.

Suppose that a subformula $r\,\mathbf{q}\,P(\bar{t})$ occurs in a formula $e\,\mathbf{q}\,A$. Then we use a term $\sigma_A^{P,f}$ to replace the term r by the term $f\bar{t}r$ in a realizer e. A term $\sigma_A^{P,f}\,e$ means a term obtained from e by replacing all such r by $f\bar{t}r$.

Note that the following (i)–(iii) hold:

(i) $FV(\sigma_A^{P,f}) \subseteq FV(A) \cup \{f\}$,

(ii) $\sigma_{A[P]}^{P,f} \equiv \sigma_{A[P']}^{P',f}$ and $\sigma_{A[P,Q]}^{P,f} \equiv \sigma_{A[P,Q']}^{P,f}$ for predicate variables Q and Q',

(iii) $\sigma_A^{P,f}\downarrow$.

4.7. Key lemma. *Let* $A[\bar{R},P]$ *be an E-formula with* $FPV(A) \subseteq \{\bar{R},P\}$. *Let* P' *be a predicate variable with* $arity(P')=arity(P)$ *and* $\bar{F},\bar{G},H,J,J'$ *be predicates with* $arity(F_i)=arity(R_i)$, $arity(G_i)=arity(R_i)+1$ (*for all* i), $arity(H)=arity(P)$, $arity(J)=arity(J')=arity(P)+1$. *We abbreviate*

$$\mathbf{q}_{\bar{R},P,P'}[\bar{F},H,H;\bar{G},J,J']$$

as $\mathbf{q}'$. *Then,*

(I) *if* P *is positive in* $A[\bar{R},P]$, *we have*

$$\lambda f.\sigma_A^{P,f}\,\mathbf{q}'\,(\forall\bar{x}(P(\bar{x})\rightarrow P'(\bar{x}))\rightarrow(A[\bar{R},P]\rightarrow A[\bar{R},P'])),$$

and

(II) *if* P *is negative in* $A[\bar{R},P]$, *we have*

$$\lambda f.\sigma_A^{P,f}\,\mathbf{q}'\,(\forall\bar{x}(P(\bar{x})\rightarrow P'(\bar{x}))\rightarrow(A[\bar{R},P']\rightarrow A[\bar{R},P])).$$

Proof. We prove the following (I′) and (II′):

(I′) If P is positive in A, we have

$$\forall f(\forall\bar{x}\forall r(H(\bar{x})\,\&\,J(r,\bar{x})\rightarrow J'(f\bar{x}r,\bar{x}))$$
$$\rightarrow\forall r(A[\bar{F},H]\,\&\,(r\,\mathbf{q}'\,A[\bar{R},P])\rightarrow\sigma_A^{P,f}r\,\mathbf{q}'\,A[\bar{R},P']))$$

(II′) If P is negative in A, we have

$$\forall f(\forall\bar{x}\forall r(H(\bar{x})\,\&\,J(r,\bar{x})\rightarrow J'(f\bar{x}r,\bar{x}))$$
$$\rightarrow\forall r(A[\bar{F},H]\,\&\,(r\,\mathbf{q}'\,A[\bar{R},P'])\rightarrow\sigma_A^{P,f}r\,\mathbf{q}'\,A[\bar{R},P])).$$

Then, (I) and (II) hold. We prove (I′) and (II′) simultaneously by induction on the complexity of A.

Case 1. Assume $A[\bar{R},P]\equiv P(\bar{t})$. We must show

$$\forall f(\forall\bar{x}\forall r(H(\bar{x})\,\&\,J(r,\bar{x})\rightarrow J'(f\bar{x}r,\bar{x}))$$
$$\rightarrow\forall r(H(\bar{t})\,\&\,J(r,\bar{t})\rightarrow J'(\sigma_A^{P,f}r,\bar{t}))).$$

But this is trivial, for $\sigma_A^{P,f}r\simeq f\bar{t}r$.

Case 2. Assume $A[\bar{R},P]$ is an atomic E-formula which is not of the form $P(\bar{t})$. This case is also trivial, since P does not occur in A.

Case 3. Assume $A\equiv A_1\,\&\,A_2$. This case is easy and left to the reader.

Case 4. Assume $A\equiv A_1\vee A_2$. This case is also easy.

Case 5. Assume $A\equiv A_1[\bar{R},P]\rightarrow A_2[\bar{R},P]$.

Subcase (i): Assume P is positive in A. Then P is negative in A_1 and positive in A_2. Suppose $\forall\bar{x}\,\forall r(H(\bar{x})\,\&\,J(r,\bar{x})\rightarrow J'(f\bar{x}r,\bar{x}))$. By the induction hypothesis,

$$\forall r(A_1[\bar{F},H]\,\&\,(r\,\mathbf{q}'\,A_1[\bar{R},P'])\rightarrow\sigma_{A_1}^{P,f}r\,\mathbf{q}'\,A_1[\bar{R},P]),\tag{1}$$

$$\forall r(A_2[\bar{F},H]\,\&\,(r\,\mathbf{q}'\,A_2[\bar{R},P])\rightarrow\sigma_{A_2}^{P,f}r\,\mathbf{q}'\,A_2[\bar{R},P']).\tag{2}$$

Suppose

$$A_1[\bar{F},H]\rightarrow A_2[\bar{F},H]\tag{3}$$

and

$$r\,\mathbf{q}'(A_1[\bar{R},P]\rightarrow A_2[\bar{R},P]).\tag{4}$$

We must show that

$$\sigma_A^{P,f}r\,\mathbf{q}'(A_1[\bar{R},P']\rightarrow A_2[\bar{R},P']),$$

that is,

$$\forall q(A_1[\bar{F},H]\,\&\,(q\,\mathbf{q}'\,A_1[\bar{R},P'])\rightarrow\sigma_A^{P,f}rq\,\mathbf{q}'\,A_2[\bar{R},P']).\tag{5}$$

Suppose

$$A_1[\bar{F},H]\tag{6}$$

and

$$q\,\mathbf{q}'\,A_1[\bar{R}, P']. \tag{7}$$

By (1),

$$\sigma_{A_1}^{P,f}\,q\,\mathbf{q}'\,A_1[\bar{R}, P]. \tag{8}$$

Since (4) means

$$\forall u(A_1[\bar{F}, H]\,\&\,(u\,\mathbf{q}'\,A_1[\bar{R}, P]) \rightarrow ru\,\mathbf{q}'\,A_2[\bar{R}, P]),$$

we have, from (6) and (8),

$$r(\sigma_{A_1}^{P,f}\,q)\,\mathbf{q}'\,A_2[\bar{R}, P]. \tag{9}$$

From (3) and (6), we get

$$A_2[\bar{F}, H]. \tag{10}$$

By (2), (9) and (10), we have

$$\sigma_{A_2}^{P,f}(r(\sigma_{A_1}^{P,f}\,q))\,\mathbf{q}'\,A_2[\bar{R}, P'].$$

But $\sigma_A^{P,f}\,rq \simeq \sigma_{A_2}^{P,f}(r(\sigma_{A_1}^{P,f}\,q))$. Hence we have got (5).

Subcase (ii): The case that P is negative in A. This subcase is proved similarly to Subcase (i).

Case 6. Assume $A \equiv \forall y A_1(y)$. This case is easy.

Case 7. Assume $A \equiv \exists y A_1(y)$. This case is also easy.

Case 8. Assume $A \equiv ((\bar{x}).A_1)(\bar{t})$. This case is also easy.

Case 9. Assume $A \equiv A[\bar{R}, P] \equiv (\mu Q.(\bar{y}).A_1[\bar{R}, P, Q])(\bar{t})$.

Subcase (i): Assume P is positive in A. Let

$$K[\bar{R}, P] \equiv \mu Q.(\bar{y}).A_1[\bar{R}, P, Q].$$

Then, we have

$$A[\bar{R}, P] \equiv K[\bar{R}, P](\bar{t})$$

and

$$K[\bar{F}, H](\bar{y}) \leftrightarrow A_1[\bar{F}, H, K[\bar{F}, H]]. \tag{11}$$

Let

$$g \equiv \mu_{m+1}\,g.\lambda\bar{y}r.\sigma_{A_1}^{Q,g}(\sigma_{A_1}^{P,f}\,r),$$

where m is the length of $\bar{y}$. Recall that $\sigma_A^{P,f} = g\bar{t}$. Suppose

$$\forall\bar{x}\forall r(H(\bar{x})\,\&\,J(r, \bar{x}) \rightarrow J'(f\bar{x}r, \bar{x})).$$

We must show

$$\forall r(K[\bar{F}, H](\bar{t})\,\&\,(r\,\mathbf{q}'\,K[\bar{R}, P](\bar{t})) \rightarrow g\bar{t}r\,\mathbf{q}'\,K[\bar{R}, P'](\bar{t})).$$

More generally, we prove

$$\forall r \forall \bar{y}(K[\bar{F},H](\bar{y}) \,\&\, (r\,\mathbf{q}'\,K[\bar{R},P](\bar{y})) \to g\bar{y}r\,\mathbf{q}'\,K[\bar{R},P'](\bar{y})).$$

This formula is equivalent to

$$\forall r \forall \bar{y}((\mu Q^*.(r,\bar{y}).(r\,\mathbf{q}'_Q[K[\bar{F},H];Q^*]\,A_1[\bar{R},P,Q]))(r,\bar{y})$$
$$\to (K[\bar{F},H](\bar{y}) \to L(g\bar{y}r,\bar{y}))), \tag{12}$$

where

$$L \equiv \mu Q^*.(r,\bar{y}).(r\,\mathbf{q}'_Q[K[\bar{F},H];Q^*]\,A_1[\bar{R},P',Q]).$$

To prove this formula using induction axiom (R2), we let

$$M \equiv (r,\bar{y}).(K[\bar{F},H](\bar{y}) \to L(g\bar{y}r,\bar{y}))$$

and prove

$$\forall r \forall \bar{y}((r\,\mathbf{q}'_Q[K[\bar{F},H];M]\,A_1[\bar{R},P,Q]) \to M(r,\bar{y})) \tag{13}$$

Now we assume

$$r\,\mathbf{q}'_Q[K[\bar{F},H];M]\,A_1[\bar{R},P,Q], \tag{14}$$

$$K[\bar{F},H](\bar{y}), \tag{15}$$

and prove

$$L(g\bar{y}r,\bar{y}).$$

By (11), we have

$$A_1[\bar{F},H,K[\bar{F},H]]. \tag{16}$$

By the induction hypothesis,

$$A_1[\bar{F},H,K[\bar{F},H]] \,\&\, (r\,\mathbf{q}'_Q[K[\bar{F},H];M]\,A_1[\bar{R},P,Q])$$
$$\to\ \sigma^{P,f}_{A_1} r\,\mathbf{q}'_Q[K[\bar{F},H];M]\,A_1[\bar{R},P',Q].$$

Hence by (14) and (16),

$$\sigma^{P,f}_{A_1} r\,\mathbf{q}'_Q[K[\bar{F},H];M]\,A_1[\bar{R},P',Q].$$

Let

$$\mathbf{q}'' \equiv \mathbf{q}'_{Q,Q'}[K[\bar{F},H],K[\bar{F},H];M,L].$$

Since Q' is not free in $A_1[\bar{R},P',Q]$, we have

$$\sigma^{P,f}_{A_1} r\,\mathbf{q}''\,A_1[\bar{R},P',Q]. \tag{17}$$

By the induction hypothesis,

$$\forall h(\forall \bar{y}\forall r(K[\bar{F},H](\bar{y})\,\&\,M(r,\bar{y})\to L(h\bar{y}r,\bar{y}))$$
$$\to\forall u(A_1[\bar{F},H,K[\bar{F},H]]\,\&\,(u\,\mathbf{q}''\,A_1[\bar{R},P',Q])\to\sigma_{A_1}^{Q;h}u\,\mathbf{q}''\,A_1[\bar{R},P',Q'])).$$

Since $\forall\bar{y}\forall r(K[\bar{F},H](\bar{y})\,\&\,M(r,\bar{y})\to L(g\bar{y}r,\bar{y}))$ holds, we have

$$\forall u(A_1[\bar{F},H,K[\bar{F},H]]\,\&\,(u\,\mathbf{q}''\,A_1[\bar{R},P',Q])\to\sigma_{A_1}^{Q;g}u\,\mathbf{q}''\,A_1[\bar{R},P'Q']).$$

Hence by (16) and (17),

$$\sigma_{A_1}^{Q;g}(\sigma_{A_1}^{P;f}r)\,\mathbf{q}''\,A_1[\bar{R},P',Q'].$$

Since Q is not free in $A_1[\bar{R},P',Q']$,

$$\sigma_{A_1}^{Q;g}(\sigma_{A_1}^{P;f}r)\,\mathbf{q}'_{Q'}[K[\bar{F},H];L]\;A_1[\bar{R},P',Q'].$$

Renaming Q' to Q,

$$\sigma_{A_1}^{Q;g}(\sigma_{A_1}^{P;f}r)\,\mathbf{q}'_{Q}[K[\bar{F},H];L]\;A_1[\bar{R},P',Q].$$

By the definition of g, we have $g\bar{y}r\simeq\sigma_{A_1}^{Q;g}(\sigma_{A_1}^{P;f}r)$. Therefore,

$$g\bar{y}r\,\mathbf{q}'_Q[K[\bar{F},H];L]\;A_1[\bar{R},P',Q].$$

By the axiom (R1),

$$(g\bar{y}r\,\mathbf{q}'_Q[K[\bar{F},H];L]\;A_1[\bar{R},P',Q])\to L(g\bar{y}r,\bar{y}).$$

Hence $L(g\bar{y}r,\bar{y})$. Thus we have derived (13).

Subcase (ii): The case that P is negative in A_1. This subcase is proved similarly to Subcase (i).

This completes the proof. □

4.8. Corollary. (i) *If P is positive in $A[\bar{R},P]$, we have*

$$\forall f(\forall\bar{x}(H(\bar{x})\to H'(\bar{x}))\,\&\,\forall\bar{x}\forall r(H(\bar{x})\,\&\,J(r,\bar{x})\to J'(f\bar{x}r,\bar{x}))$$
$$\to\forall r(A[\bar{F},H]\,\&\,(r\,\mathbf{q}''\,A[\bar{R},P])\to\sigma_A^{P;f}r\,\mathbf{q}''\,A[\bar{R},P'])),$$

where $\mathbf{q}''$ is $\mathbf{q}_{\bar{R},P,P'}[\bar{F},H,H';\bar{G},J,J']$. That is,

$$\lambda f.\sigma_A^{P;f}\,\mathbf{q}''\,(\forall\bar{x}.(P(\bar{x})\to P'(\bar{x}))\to(A[\bar{R},P]\to A[\bar{R},P'])).$$

(ii) *Let*

$$D\equiv(r,\bar{x}).(B(\bar{x})\to f\bar{x}r\;\mathbf{q}\;C(\bar{x})).$$

If P is positive in $A[P]$, we have

$$\forall\bar{x}(B(\bar{x})\to C(\bar{x}))\to\forall r(A[B]\,\&\,(r\,\mathbf{q}_P[B;D]\;A[P])\to\sigma_A^{P;f}r\;\mathbf{q}\;A[C]).$$

Proof. (i) Suppose $\forall\bar{x}(H(\bar{x})\to H'(\bar{x}))$. By the above lemma.

$$\forall f(\forall\bar{x}\forall r(H(\bar{x})\,\&\,J(r,\bar{x})\to J'(f\bar{x}r,\bar{x}))\to\forall r(A[\bar{F},H]\,\&\,(r\,\mathbf{q}'\,A[\bar{R},P])$$
$$\to\sigma_A^{P,f}r\,\mathbf{q}'\,A[\bar{R},P'])),$$

where $\mathbf{q}'$ is $\mathbf{q}_{\bar{R},P,P'}[\bar{F},H;\bar{G},J,J']$. Since P' does not occur in $A[\bar{R},P]$,

$$(r\,\mathbf{q}'\,A[\bar{R},P])\leftrightarrow(r\,\mathbf{q}''\,A[\bar{R},P]).$$

Moreover, by Lemma 4.2,

$$(\sigma_A^{P,f}r\,\mathbf{q}'\,A[\bar{R},P'])\to\sigma_A^{P,f}r\,\mathbf{q}''\,A[\bar{R},P'].$$

Hence (i) holds.

(ii) Suppose $\forall\bar{x}(B(\bar{x})\to C(\bar{x}))$. Let

$$\mathbf{q}''\equiv\mathbf{q}_{P,P'}[B,C;D,(r,\bar{x}).(r\,\mathbf{q}\,C(\bar{x}))].$$

By (i),

$$\forall h(\forall\bar{x}\forall r(B(\bar{x})\,\&\,D(r,\bar{x})\to h\bar{x}r\,\mathbf{q}\,C(\bar{x}))\to\forall r(A[B]\,\&\,(r\,\mathbf{q}''\,A[P])$$
$$\to\sigma_A^{P,h}r\,\mathbf{q}''\,A[P'])).$$

Since $\forall\bar{x}\forall r(B(\bar{x})\,\&\,D(r,\bar{x})\to f\bar{x}r\,\mathbf{q}\,C(\bar{x}))$ holds, we have

$$\forall r(A[B]\,\&\,(r\,\mathbf{q}''\,A[P])\to\sigma_A^{P,f}r\,\mathbf{q}''\,A[P']).$$

Since P' is not free in $A[P]$ and P is not free in $A[P']$,

$$\forall r(A[B]\,\&\,(r\,\mathbf{q}_P[B;D]\,A[P])\to\sigma_A^{P,f}r\,\mathbf{q}_{P'}[C;(r,\bar{x}).(r\,\mathbf{q}\,C(\bar{x}))]\,A[P']).$$

By Lemma 4.3,

$$\forall r(A[B]\,\&\,(r\,\mathbf{q}_P[B;D]\,A[P])\to\sigma_A^{P,f}r\,\mathbf{q}\,A[C]).\qquad\square$$

Now let us construct a realizer of the induction axiom (R2).

4.9. Theorem. *Let* $B\equiv\mu P.(\bar{x}).A[P]$. *Then*

$$\lambda q.\mu f.\lambda\bar{x}r.q\bar{x}(\sigma_A^{P,f}r)\,\mathbf{q}\,(\forall\bar{x}(A[C]\to C(\bar{x}))\to\forall\bar{x}(B(\bar{x})\to C(\bar{x})))$$

holds.

Proof. Suppose

$$\forall\bar{x}(A[C]\to C(\bar{x}))\tag{18}$$

and

$$q\,\mathbf{q}\,\forall\bar{x}(A[C]\to C(\bar{x})).\tag{19}$$

Let $f \equiv \mu f.\lambda \bar{x} r.q\bar{x}(\sigma_A^{P,f} r)$. We show that

$$\forall r, \bar{x}(B(\bar{x}) \,\&\, (r \,\mathbf{q}\, B(\bar{x})) \to f\bar{x}r \,\mathbf{q}\, C(\bar{x})) \,\&\, \forall \bar{x}(f\bar{x}\downarrow). \tag{20}$$

$\forall \bar{x}(f\bar{x}\downarrow)$ is trivial. The remaining part is equivalent to

$$\forall r, \bar{x}((\mu P^*.(r, \bar{x}).(r \,\mathbf{q}_P[B; P^*]\, A[P]))(x, \bar{x}) \to (B(\bar{x}) \to f\bar{x}r \,\mathbf{q}\, C(\bar{x}))).$$

So, by (R2), it is sufficient to prove that

$$(r \,\mathbf{q}_P[B; D]\, A[P]) \to D(r, \bar{x}) \tag{21}$$

where $D \equiv (r, \bar{x}).(B(\bar{x}) \to f\bar{x}r \,\mathbf{q}\, C(\bar{x}))$. Suppose

$$(r \,\mathbf{q}_P[B; D]\, A[P]) \,\&\, B(\bar{x}).$$

Then, since $B(\bar{x}) \leftrightarrow A[B]$, we get $A[B]$. From (18) and (R2) we have

$$\forall \bar{x}[B(\bar{x}) \to C(\bar{x})). \tag{22}$$

By Corollary 4.8 (ii), we have $\sigma_A^{P,f} r \,\mathbf{q}\, A[C]$. Since P is positive in $A[P]$, (22) implies $A[B] \to A[C]$. So we have $A[C]$. Then, by (19), $q\bar{x}(\sigma_A^{P,f} r) \,\mathbf{q}\, C(\bar{x})$. By the definition of f, $f\bar{x}r \simeq q\bar{x}(\sigma_A^{P,f} r)$ holds. Therefore we get $f\bar{x}r \,\mathbf{q}\, C(\bar{x})$. Thus we have (21). Then the theorem holds. □

4.10. Theorem (Soundness of realizability). *If* $\mathbf{EON} + \mu \vdash A$ *holds, then we can get a term e effectively from the proof and* $\mathbf{EON} + \mu \vdash e \,\mathbf{q}\, A$ *holds.*

Proof. The soundness for the axioms and rules of **EON** is proved in [1].

The axiom (A1) is realized by Lemma 4.1. The axiom (R1) is realized by Corollary 4.5. The axiom (R2) is realized by Theorem 4.9. □

This theorem improves the result of [10]. In [10], the system can only use $\mu P.\bar{x}.A[P]$ under the following conditions: each occurrence of a predicate variable P in an E-formula $A[P]$ must be (1) positive and (2) either strictly positive in $A[P]$ or in some Harrop subformula of $A[P]$. On the other hand, in our system the condition (2) can be omitted and our system can use $\mu P.\bar{x}.A[P]$ under the condition (1). In particular, our system covers nonstrictly positive inductive definitions, which the system in [10] cannot cover.

4.11. Corollary. (1) *The term existence property holds for* $\mathbf{EON} + \mu$. *That is, if* $\mathbf{EON} + \mu \vdash \exists x A(x)$ *holds, we can get a term t effectively from the proof and* $\mathbf{EON} + \mu \vdash A(t)$ *holds.*

(2) *The disjunction property holds for* $\mathbf{EON} + \mu$. *That is, if* $\mathbf{EON} + \mu \vdash A \vee B$ *holds, then* $\mathbf{EON} + \mu \vdash A$ *holds or* $\mathbf{EON} + \mu \vdash B$ *holds.*

These claims are proved easily by using the soundness theorem.

4.12. Theorem (Program extraction). *Suppose that* $\mathbf{EON}+\mu \vdash A(x) \to (jx\ \mathbf{q}\ A(x))$. *If* $\mathbf{EON}+\mu \vdash \forall x(A(x) \to \exists y B(x,y))$ *holds, we can get a term* f *effectively from the proof and* $\mathbf{EON}+\mu \vdash \forall x(A(x) \to fx\downarrow \mathbin{\&} B(x,fx))$ *holds.*

Proof. By the soundness theorem and $\mathbf{EON}+\mu \vdash \forall x(A(x) \to \exists y B(x,y))$, we can get a term e from the proof and $\mathbf{EON}+\mu \vdash (e\,\mathbf{q}\,\forall x(A(x) \to \exists y B(x,y)))$ holds. Put $f \equiv \mathbf{p}_0(ex(jx))$. Then the claim holds. $\square$

By the program extraction theorem, we can extract a program f from the constructive proof of the formula $\forall x(A(x) \to \exists y B(x,y))$ in $\mathbf{EON}+\mu$.

Recently some meaningful examples of programs use nonstrictly positive inductive definitions. Hoffmann found an example of breadth first search of trees, which uses continuations. The type of continuations is given by a nonstrictly positive datatype *cont*, which is defined by $cont = \mathrm{D}\,|\,\mathrm{C}$ of $((cont \to \alpha) \to \alpha)$. In our system, the predicate $C(x)$ which states that x is a continuation can be described as follows:

$$C \equiv \mu P.\lambda x.x = \mathrm{D} \vee \exists y(x = \mathrm{C}(y) \mathbin{\&} \forall f(\forall z(P(z) \to A(f(z))) \to A(y(f))))$$

for some predicate $A(x)$.

When $A(x)$ is not Harrop, the above inductive definition cannot be used in the system of [10], but our system can use the above inductive definition.

5. Concluding remarks

In this paper, we treated Kleene-style $\mathbf{q}$-realizability. The results of this paper can be applied to Grayson-style $\mathbf{q}$-realizability by small modification. The proof of the soundness theorem for Grayson-style realizability will be simpler than our proof for Kleene-style realizability.

The results of this paper can be also applied to usual $\mathbf{r}$-realizability by small modification. The soundness theorem for $\mathbf{r}$-realizability will give fundamental tools to prove choice principles, consistency and conservative extension results in the area of mathematical logic.

References

[1] M.J. Beeson, *Foundations of Constructive Mathematics* (Springer, Berlin, 1985).
[2] M. Beeson, Proving programs and programming proofs, in: R. Barcan Marcus, G.J.W. Dorn and P. Weingarter, eds., *Logic, Methodology, and Philosophy of Science VII* (North-Holland, Amsterdam, 1985) 51–82.
[3] W. Buchholz, S. Feferman, W. Pohlers and W. Sieg, *Iterated Inductive Definitions and Subsystems of Analysis: Recent Proof-Theoretical Studies*, Lecture Notes in Math. Vol. 897 (Springer, Berlin, 1981).
[4] R.L. Constable et al., *Implementing Mathematics with the Nuprl Proof Development System* (Prentice-Hall, Englewood Cliffs, NJ, 1986).

[5] S. Feferman, A language and axioms for explicit mathematics, Lecture Notes in Math. Vol. 450 (Springer, Berlin, 1975) 87–139.
[6] S. Feferman, Constructive theories of functions and classes, in: M. Boffa, D. van Dalen and K. McAloon, eds., *Logic Colloquium '78, Proceedings of the Logic Colloquium at Mons* (North-Holland, Amsterdam, 1979) 159–224.
[7] S. Hayashi and H. Nakano, **PX**: *A Computational Logic* (MIT Press, Cambridge, MA, 1988).
[8] P. Martin-Löf, Constructive mathematics and computer programming, in: L.J. Cohen et al., eds., *Logic, Methodology, and Philosophy of Science VI* (North-Holland, Amsterdam, 1982) 153–179.
[9] B. Nordström, K. Petersson and J.M. Smith, *Programming in Martin-Löf's Type Theory* (Oxford University Press, Oxford, 1990).
[10] M. Tatsuta, Program synthesis using realizability, *Theoret. Comput. Sci.*, **90** (1991) 309–353.
[11] A.S. Troelstra and D. van Dalen, *Constructivism in Mathematics, Vols. I, II* (North-Holland, Amsterdam, 1988).